Eidgenössische Technische Hochschule Zürich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Prof. Petros Koumoutsakos
CSE Lab, ETH Zürich
www.cse-lab.ethz.ch

Advanced Computational Science
SS 2009

Serie 4 Ausgabe: March 17, 2009 Abgabe: March 24, 2009

Guidelines

▷ Submission: Submit a written or typed report with your answers and graphics.

▷ Code: Send your source code (and makefile) to the TA of your session. Use "[ACS] Homework4" in the subject of your email.

▷ Credit: Your report should list all the contributors.

▷ Bugs: Print-outs of your source code are *not* required in your report, unless you have bugs. Help us give you partial marks.

▷ Deadline: Assignments have to be handed in at the beginning of the next exercise session.

Exercise 1 (40 points) Adaptive Quadrature

For many functions the subdivision process of $[a, b]$ into equally spaced subintervals is not optimal since small subintervals are not necessary in regions where the integrand is smooth.

An alternative is adaptive quadrature, a technique in which each subinterval $[x_i, x_{i+1}]$ is refined only if it is determined that the quadrature rule has not computed the subinterval

$$I_i = \int_{x_i}^{x_{i+1}} f(x)\, dx \tag{1}$$

with sufficient accuracy.

We will use the Simpson quadrature rule. The error is given by

$$S_0^h(I_i) = I_i + c_4 h^4 + O(h^6)\ . \tag{2}$$

By applying Richardson extrapolation to two successive Simpson approximations S_0^h and $S_0^{h/2}$, the error can be approximated by

$$\left| I_i - S_0^h \right| \approx \left| S_1^{h/2} - S_0^h \right| =: \delta \tag{3}$$

where S^1 is the Richardson-extrapolated approximation.

Each subinterval is refined only if $\delta > \epsilon$, where ϵ is a user defined constant.

▷ Part 1 (20 points) Develop and implement a recursive algorithm performing adaptive quadrature with the Simpson rule.
Your algorithm has to be efficient. The main objective of an adaptive quadrature is the minimization of the number of function evaluations! These function evaluations can be expensive in real life cases: it can be the result of an expensive numerical simulation, or the result of a time-consuming experiment.
Your code should take as input the function to evaluate and the threshold ϵ. The output should consist of the quadrature result and the number of function evaluations.

▷ Part 2 (20 Points) Approximate the integral

$$\int_{-1}^{1} 10e^{-500x^2} - \frac{0.01}{(x+0.5)^2+0.001} + 5\cos(5x)dx \tag{4}$$

with your adaptive quadrature code for $\epsilon = 10^{-9}, 10^{-8.5}, 10^{-8}, \ldots, 10^{-2}$ and plot the error. Demonstrate the adaptivity visually (for a large value of ϵ) by means of a scatter plot of the (x, y) points used for the quadrature. Comment your graphic.
Compare the adaptive quadrature procedure to a non-adaptive composite Simpson rule (already developed in a previous homework): plot the logarithm of the error versus the logarithm of the number of function evaluations used for both schemes in a single figure.
Comment your results.